AD-757 029

ANTISUBMARINE ATTACK: COMPUTER PROGRAM 13-64P

S. A. Denenberg, et al

Center for Naval Analyses
Arlington, Virginia

28 August 1964

AD757029

CNA

ANTISUBMARINE ATTACK: COMPUTER PROGRAM 13-64P

By S.A. Denenberg and
A. Hershaft

Research Contribution No. 60

CONTRACT NONR 3732(00)

Research Contribution

OPERATIONS EVALUATION GROUP

Center for Naval Analyses

WASHINGTON 25 D C

28 August 1964

RESEARCH CONTRIBUTION

Operations Evaluation Group

CENTER FOR NAVAL ANALYSES

ANTISUBMARINE ATTACK: COMPUTER PROGRAM 13-64P

By S. A. Denenberg and
A. Hershaft

Research Contribution No. 60

S. A. Denenberg

A. Hershaft

This Research Contribution does not necessarily represent the views of CNA or the U.S. Navy. It may be modified or withdrawn at any time.

I-a.

ABSTRACT

An IBM 7090 computer program is described which calculates the distribution of distances between the point of activation of a weapon and a moving target submarine taking into account the estimated component attack errors. The model used is more flexible and realistic than similar past efforts and is expected to produce more reliable submarine kill probabilities. The miss distances are computed by Monte Carlo simulation of the actual tracking and firing tactics. They are plotted by an SC 4020 plotter, first in ascending order, then as a cumulative frequency distribution. Flow charts, a listing of the FORTRAN program, and a sample calculation are included.

TABLE OF CONTENTS

I. INTRODUCTION

The determination of the probability of killing an evading submarine has long been a major problem of antisubmarine warfare. Past attack models have been usually restricted to specific tactical situations or sensor/weapon systems and required rather awkward assumptions about the nature of distribution of the various attack component errors.

The effects of antisubmarine weapons as a function of distance are comparatively well known. Therefore, the major difficulty in the determination of kill probabilities lies in the calculation of the probability of placing a weapon within a given distance from the target, in terms of the expected errors in the various stages of attack.

This program provides a distribution of these distances by Monte Carlo simulation of the actual antisubmarine tracking and firing tactics. It is based on a model (reference (a)) which is both flexible enough to cover most tactical situations and sensor/weapon systems and realistic enough to yield reliable kill probability values. The use of the Monte Carlo technique voids the need for unreasonable distortion of component error distributions.

II. GENERAL DESCRIPTION

Model

The tactical situation is shown in figure 1. The attack unit is at the origin of the coordinate system. The true course of the submarine is parallel to the x-axis. At the time $t_1 = 0$, the submarine is at the true point P_1 with polar coordinates (r_1, α)*. The submarine proceeds from P_1 at a speed U_1 and arrives at the point P_2 at time t_2. It then executes a turn of radius R_t thru γ degrees of arc at a velocity U_2. When the turn thru γ is completed at time t_3 at point P_3, the submarine continues on a tangential course at speed U_3, reaching point P_4 at a depth Z at time t_4.

The time t_4 is set equal to the actual activation** time of the attack weapon, thus making P_4 a variable position dependent on t_4. If t_4 is less than t_2,

* All angles are measured from the vertical or the range line, positive direction to the right. Deflections are measured from the range line, positive to the right.

** Activation here denotes the step in which the weapon begins to exert its effect on the target (e.g., beginning of search for a homing torpedo or detonation for a depth charge).

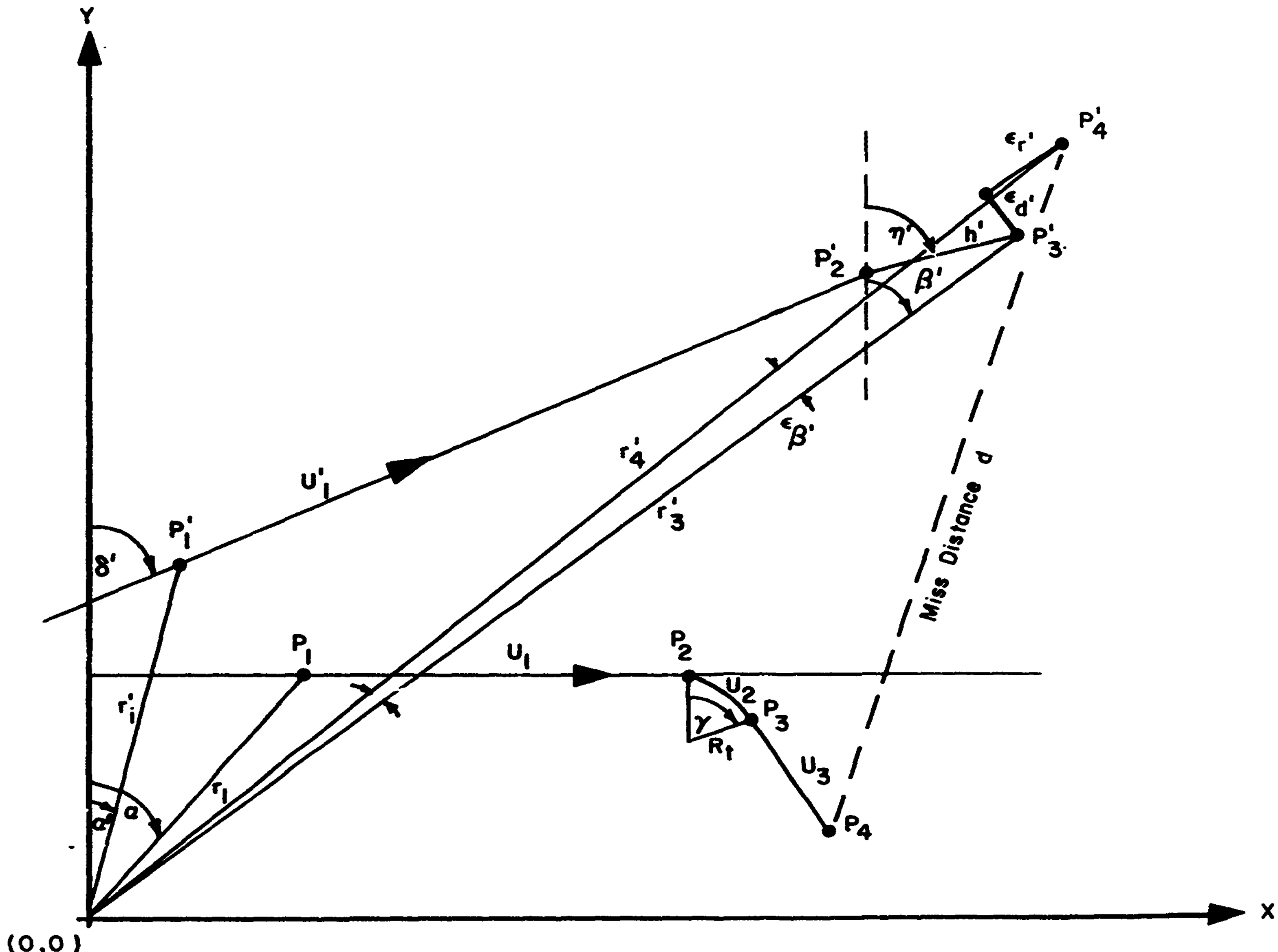

Y
X
(0,0)
P'_4
ϵ_r'
ϵ_d'
P'_3
h'
η'
P'_2
β'
$\epsilon_{\beta'}$
r'_4
r'_3
U'_1
P'_1
δ'
r'_i
P_1
U_1
P_2
U_2
P_3
γ
R_t
U_3
P_4
Miss Distance d
r_1
α
α'

FIGURE 1

P_4 will be somewhere between P_1 and P_2. If t_4 is greater than t_2 but less than t_3, P_4 will be somewhere on the circular arc between P_2 and P_3. Finally, if t_4 is greater than t_3, P_4 will be located somewhere along the tangent to the circular arc depending on the time elapsed between t_3 and t_4.

Due to errors in tracking, the attack unit presumes the submarine to be at the point P'_1 with the polar coordinates (r'_1, α') at time $t'_1 = t_1 = 0$*. The attack unit loses contact with the submarine at P'_1 where it has measured the submarine's speed as U'_1 and its course as δ'. The attacker assumes that the submarine continues from P'_1 at the speed U'_1 on the course δ' and calculates P'_2, where P'_2 is the assumed location of the submarine at t'_3, the desired time of activation of the weapon.

The aimpoint of the weapon P'_3 is specified by an offset angle η' and an offset distance h' from the predicted point P'_2. Due to weapon placement errors, the weapon arrives at the point P'_4 at depth Z', and activates at time t'_4. The time t_4 is set equal to t'_4 so that the distance between P_4 and P'_4 becomes the miss distance between the submarine and the weapon.

Program

A miss distance d_i is calculated for each Monte Carlo iteration. At the end of the iteration process, two graphs are plotted:

Graph 1: The miss distances d_i are plotted in sequential ascending order.

Graph 2: The cumulative frequency distribution of the miss distances is plotted as follows: D_M is an input to the program and is used to specify the largest value on the x-axis. If $D_M = 0$ or is not specified, the largest d_i or d_{max} is found.

D_M is then chosen as the smallest number of the form 1×10^n, 2×10^n, 5×10^n, $-2 \leq n \leq 38$, which is greater than d_{max}. Once D_M has been chosen, the x-axis of Graph 2 is divided into N_s equal increments, where N_s is also an input to the program.

* Primed variables are used to describe quantities in the observed system. Unprimed variables denote quantities in the true or actual system.

Each increment represents a range of miss distances. The frequency of occurrence for each range of miss distance is calculated and the cumulative frequencies are plotted.

The largest and smallest values of d_i and the number $d_i > D_M$ are printed out. All values of $d_i > D_M$ are ignored; they are not plotted on Graph 1 and they are not used in determining the cumulative frequency distribution of Graph 2.

III. METHOD OF SOLUTION

The inputs for the program consist of the following parameters:

Address	Symbol	Description
1	ID	Identification number which can be assigned to a computer run and which will appear on the graphs and the printed output. ID is an integer such that $0 \le ID \le 32,767$
2	N_I	Number of Monte Carlo iterations.
3	D_M	The value of miss distance that will be used as the largest value on the x-axis of Graph 2 (feet)*
4	N_S	Number of equal increments on the x-axis of the cumulative frequency plot (one increment has length = D_M/N_S**)
5	N_E	Number of "empty" passes thru random number generators.***
6	r_1	Distance from attack unit to P_1, the submarine's true position when the last contact is made (feet)
7	Δr_1	Observed position bias error in range (feet)
8	σr_1	One standard deviation of position error in range (feet)

* If not entered, D_M is chosen as the smallest number of the form 1×10^n, 2×10^n, 5×10^n $-2 \le n \le 38$ which is greater than the largest calculated miss distance.

** If not entered, N_S will be set equal to 500.

*** N_E passes thru the random number generators are made only for the first data set processed by the program.

Address	Symbol	Description
9	r_t	Radius of turn that submarine executes at P_2 (yards)
10	α	True bearing from attack unit to submarine at point of last contact (degrees)
11	$\Delta\alpha$	Observed position bias error in bearing (degrees)
12	σ_α	One standard deviation of position error in bearing (degrees)
13	γ	Number of degrees of arc in turn of radius r_t.
14	σ_γ	__If positive,__ σ_γ is one standard deviation of error in the turn angle γ (degrees). γ will be replaced by a normal distribution about γ. __If negative,__ γ will be replaced by a uniform distribution from 0 to γ. __If zero,__ γ will not be changed.
15	$\Delta\delta$	Observed course bias error, where true course $\delta = 90$ (degrees)
16	σ_δ	One standard deviation of course error (degrees)
17	u_1	True speed of the submarine from P_1 to P_2 (knots)
18	u_2	True speed of the submarine from P_2 to P_3 (knots)
19	u_3	True speed of the submarine from P_3 to P_4 (knots)
20	Δu_1	Observed bias error of submarine speed u_1 (knots)
21	σ_{u_1}	One standard deviation of error in submarine speed u_1 (knots)
22	t_2	True time at which submarine starts turn (seconds)
23	σ_{t_2}	__If positive,__ σ_{t_2} is one standard deviation of error in the time t_2 (seconds). t_2 will be replaced by a normal distribution about t_2. __If negative,__ t_2 will be replaced by a uniform distribution from 0 to t_2. __If zero,__ t_2 will not be changed.
24	t'_3	Desired time of activation of the weapon (seconds)
25	η'	Aimpoint offset angle (degrees)

Address	Symbol	Description
26	h'	Aimpoint offset distance (feet)
27	$\Delta\epsilon_{r'}$	Weapon placement bias error in range (feet)
28	$\sigma_{\epsilon_{r'}}$	One standard deviation of weapon placement error in range (feet)
29	$\Delta\epsilon_{d'}$	Weapon placement bias error in lateral displacement (feet)*
30	$\sigma_{\epsilon_{d'}}$	One standard deviation of weapon placement error in lateral displacement (feet)*
31	$\Delta\epsilon_{\beta'}$	Weapon placement bias error in bearing (degrees)*
32	$\sigma_{\epsilon_{\beta'}}$	One standard deviation of weapon placement error in bearing (degrees)*
33	Z	True depth of submarine when weapon is activated (feet)
34	ΔZ	If σ_Z is positive, ΔZ is considered to be the observed depth bias error (feet). If σ_Z is negative, ΔZ is considered to be the maximum operating depth of the evading submarine (feet).
35	σ_Z	If positive, σ_Z is one standard deviation of error in the depth Z (feet). The observed depth of the submarine, Z', will be calculated as a normal distribution about $Z + \Delta Z$ (feet). If negative, Z' will be calculated as a uniform distribution from 0 to ΔZ, the maximum operating depth of the submarine (feet). If zero, Z' will be set equal to Z (feet).
36	v'	Velocity of the weapon (feet/seconds).

In the following discussion, R_N is a Gaussian-distributed (mean = 0, standard deviation = 1) random number which is always less than 4 standard deviations. R_U is a uniformly-distributed (mean = 0) random number.

* If $\Delta\epsilon_{d'} = \sigma_{\epsilon_{d'}} = 0$, then Option 2 (see page 8) will be used.

If $\Delta\epsilon_{\beta'} = \sigma_{\epsilon_{\beta'}} = 0$, then Option 1 (see page 8) will be used.

The miss distance d will be computed once the x-y coordinates of P_4 and P'_4 are determined.

1. It is first necessary to calculate the x and y coordinates of P'_3. In order to do this, certain variables in the observed system must be computed.

Since r_t, Δu_1, σu_1, u_1, u_2, and u_3 will be used in subsequent calculations, they must be scaled to be consistent with the units of the rest of the input parameters.

$$U_i = 1.6878065 u_i, \quad i = 1, 2, 3$$

$$\Delta U_1 = 1.6878065 \Delta u_1$$

$$\sigma_{U_1} = 1.6878065 \sigma_{u_1}$$

$$R_t = 3 r_t$$

Then $$U'_1 = U_1 + \Delta U_1 + R_{N_1} \sigma_{U_1}$$

$$\alpha' = \alpha + \Delta\alpha + R_{N_2} \sigma_\alpha$$

$$r'_1 = r_1 + \Delta r_1 + R_{N_3} \sigma_{r_1}$$

$$\delta' = 90 + \Delta\delta + R_{N_4} \sigma_\delta \quad (\delta = 90^\circ \text{ in the true system})$$

It is now possible to calculate the x and y coordinates of P'_3, x'_3 and y'_3, from the geometry of figure 1.

$$x'_3 = r'_1 \sin\alpha' + U'_1 t'_3 \sin\delta' + h' \sin\eta'$$

$$y'_3 = r'_1 \cos\alpha' + U'_1 t'_3 \cos\delta' + h' \cos\eta'$$

2. The next step is the determination of x'_4 and y'_4, the x and y coordinates of P'_4. This calculation depends on how the weapon placement error is described:

Option 1: The weapon placement error is described by range and deflection (ϵ_r, and ϵ_d, respectively)

Option 2: The weapon placement error is described by range and bearing (ϵ_r, and ϵ_β, respectively)

Option 1

$$\epsilon_{r'} = \Delta\epsilon_{r'} + R_{N_5}\,\sigma_{\epsilon_{r'}}$$

$$\epsilon_{d'} = \Delta\epsilon_{d'} + R_{N_6}\,\sigma_{\epsilon_{d'}}$$

($\epsilon_{d'}$ is measured positive to the right along the range line from the attack unit to the submarine)

$$r'_3 = \sqrt{(x'_3)^2 + (y'_3)^2}$$

$$x'_4 = x'_3 + \epsilon_{r'}\frac{x'_3}{r'_3} + \epsilon_{d'}\frac{y'_3}{r'_3} \quad *$$

$$y'_4 = y'_3 + \epsilon_{r'}\frac{y'_3}{r'_3} - \epsilon_{d'}\frac{x'_3}{r'_3} \quad *$$

$$r'_4 = \sqrt{(x'_4)^2 + (y'_4)^2}$$

Option 2

$$\epsilon_{r'} = \Delta\epsilon_{r'} + R_{N_5}\,\sigma_{\epsilon_{r'}}$$

$$\epsilon_{\beta'} = \Delta\epsilon_{\beta'} + R_{N_6}\,\sigma_{\epsilon_{\beta'}}$$

$$r'_3 = \sqrt{(x'_3)^2 + (y'_3)^2}$$

$$r'_4 = r'_3 + \epsilon_{r'}$$

$$\beta' = \arctan\frac{x'_3}{y'_3}$$

$$x'_4 = r'_4 \sin(\beta' + \epsilon_{\beta'})$$

$$y'_4 = r'_4 \cos(\beta' + \epsilon_{\beta'})$$

---

* The derivation of these equations is somewhat lengthy and is therefore not included.

The time of activation of the weapon is calculated taking into account the delay due to weapon placement error.

$$t'_4 = t'_3 + \frac{r'_4 - r'_3}{v'}$$

3. The x and y coordinates of P_4, x_4 and y_4, must now be determined:

Since t_2 and γ will be used in subsequent calculations, they must first be transformed in accordance with the convention defined on page 5.

$$\left.\begin{array}{l} \text{If } \sigma_{t_2} > 0, \text{ then } T_2 = t_2 + R_{N_7}\,\sigma_{t_2} \\ \text{If } \sigma_{t_2} < 0, \text{ then } T_2 = R_{U_1}\,t_2 \\ \text{If } \sigma_{t_2} = 0, \text{ then } T_2 = t_2 \end{array}\right\}$$

$$\left.\begin{array}{l} \text{If } \sigma_\gamma > 0, \text{ then } \Gamma = \gamma + R_{N_8}\,\sigma_\gamma \\ \text{If } \sigma_\gamma < 0, \text{ then } \Gamma = R_{U_2}\,\gamma \\ \text{If } \sigma_\gamma = 0, \text{ then } \Gamma = \gamma \end{array}\right\}$$

Since T_2 and Γ have now been determined, t_3 is given by:

$$t_3 = T_2 + \frac{\pi}{180}\,\Gamma\,\frac{R_t}{U_2}$$

As was noted previously, the location of P_4 is dependent upon t'_4; thus, there exist three possibilities:

a) $t'_4 \le T_2$

then $x_4 = r_1 \sin\alpha + U_1 t'_4$

$y_4 = r_1 \cos\alpha$

b) $T_2 < t'_4 \leq t_3$

then let $\theta = \left(\frac{t'_4 - T_2}{t_3 - T_2}\right) \Gamma$

$$x_4 = r_1 \sin\alpha + U_1 T_2 + R_t \sin\theta$$

$$y_4 = r_1 \cos\alpha - R_t (1 - \cos\theta)$$

c) $t'_4 > t_3$

then let $S = U_3 (t'_4 - t_3)$

$$x_4 = r_1 \sin\alpha + U_1 T_2 + R_t \sin\gamma + S \cos\gamma$$

$$y_4 = r_1 \cos\alpha - R_t (1 - \cos\gamma) - S \sin\gamma$$

4. The final step is the calculation of the miss distance:

If $\sigma_Z > 0$, then $Z' = Z + \Delta Z + R_{N_9} \sigma_Z$ *

If $\sigma_Z < 0$, then $Z' = R_{U_3} \Delta Z$

If $\sigma_Z = 0$, then $Z' = Z$

Then $d = \sqrt{(x_4 - x'_4)^2 + (y_4 - y'_4)^2 + (Z - Z')^2}$

* See page 6 to review the definition of σ_Z.

IV. USER'S INSTRUCTIONS

See the list of input parameters on page 4. The user should submit a data submittal form (see appendix C). These forms have space for the submitter's name, the date, the program number (13-64P), the security classification if any, the parameter addresses and values, and an estimate of the computer running time (see section VIII).

Input flexibility has been attained by allowing the user to vary any or all of the parameters in a computer run. In any one set of data the parameters remain fixed, of course, but there is no programmed limit to the number of data sets a user may submit in a run. The only restriction is that each data set must terminate with one blank card, and the last set in the run must terminate with two blank cards.

A further advantage enjoyed by the user is that for each data set after the first, he need submit only those parameter values in a set that are different from those in the previous set.* This is accomplished by identifying each parameter by a unique "address" in the computer memory (see appendix E). Initially every address is cleared to zero so that only non zero input parameters need be entered.

V. SAMPLE PROBLEM

A situation was fabricated for illustrative purposes whereby most of the bias errors were chosen as one percent of the true value and standard deviations were taken as ten percent of the true value. The output is shown in appendix D; and the problem submittal form in appendix C.

VI. KEYPUNCH INSTRUCTIONS

See the list of input parameters on page 4 and the sample problem submittal form in appendix C. Input card formats are described in appendix E.

VII. OPERATOR'S INSTRUCTIONS

Run under control of the Bell System on the IBM 7090 computer. Graphs are made by the Stromberg-Carlson microfilm recorder. A charactron (CRT) tape must be mounted on tape unit A8 in the low-density mode. When the program is run at the David Taylor Model Basin all that is necessary for plotting is a "BCRT" control card (columns 8 thru 12) and the "AMPLOS" subroutine. When the program ~~is~~ run at CEIR, NAVIC or NAVCOSSACT, the "BCRT" control card is not to be used and the "AMPLOS" subroutine is to be replaced by the "CXPLOT" and "AMPLOF" subroutines; the CRT tape on A8 can then be taken to DTMB or Stromberg-Carlson for processing.

* The exception to this rule is N_E, the number of empty passes through the random number generator (see page 4).

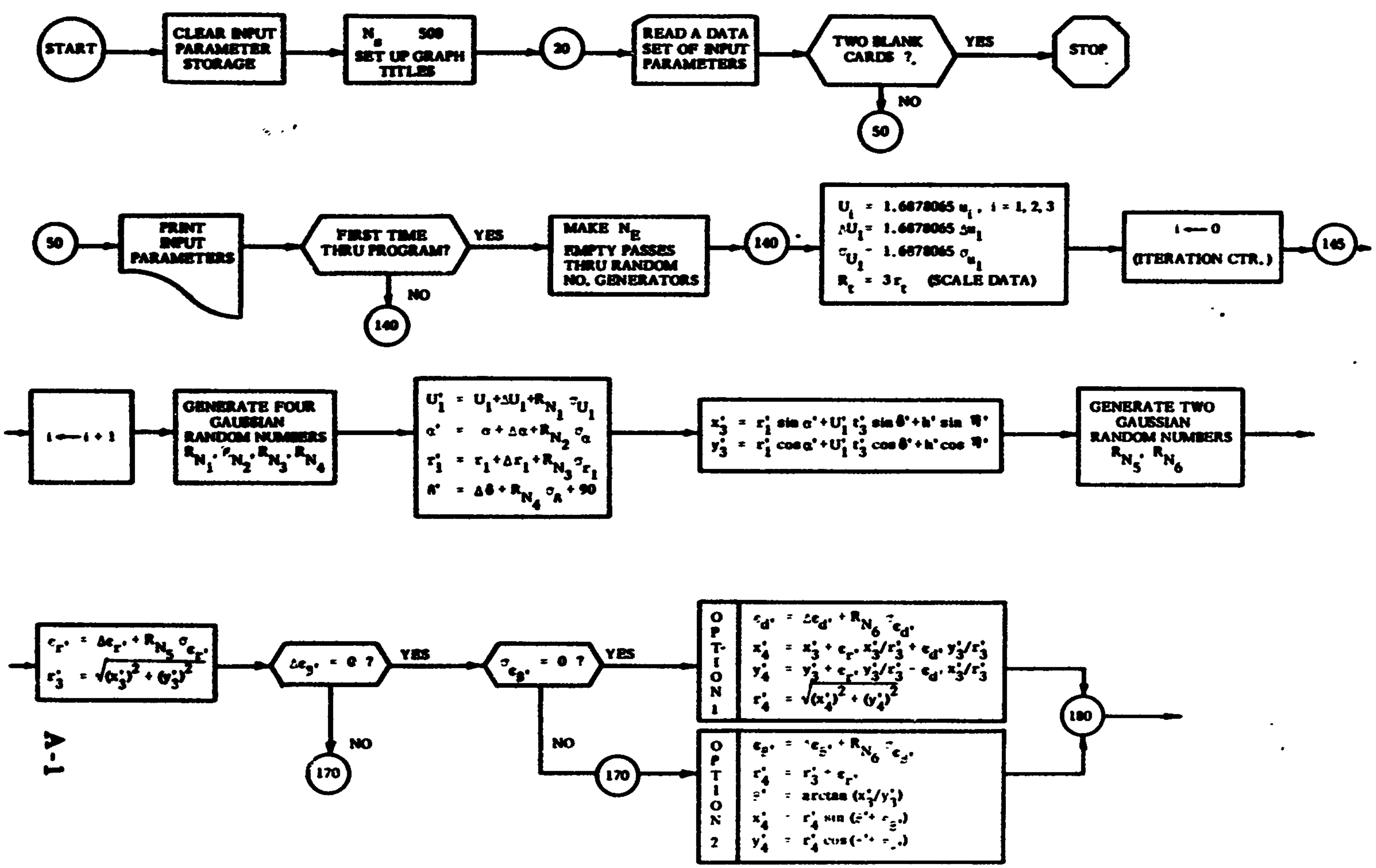
VIII. TIMING

The table presented below gives a conservative estimate of the computer running time for one data set as a function of N_I, the number of Monte Carlo iterations.

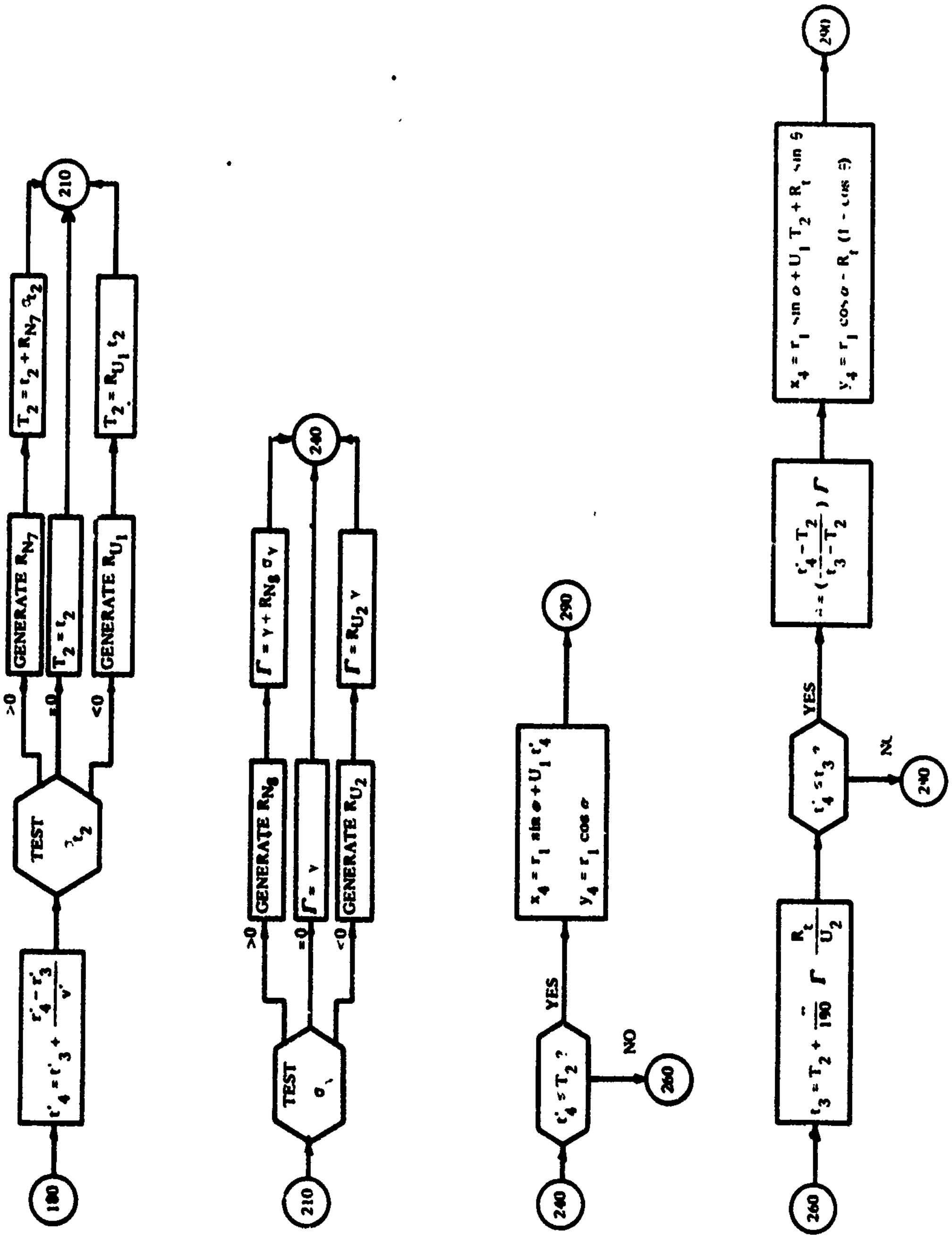
N_I (Number of Monte-Carlo Iterations)	Computer Running Time (in minutes)
100	2
1000	2
2000	3
3000	3
4000	4
5000	5
6000	5
7000	6
8000	6
9000	6

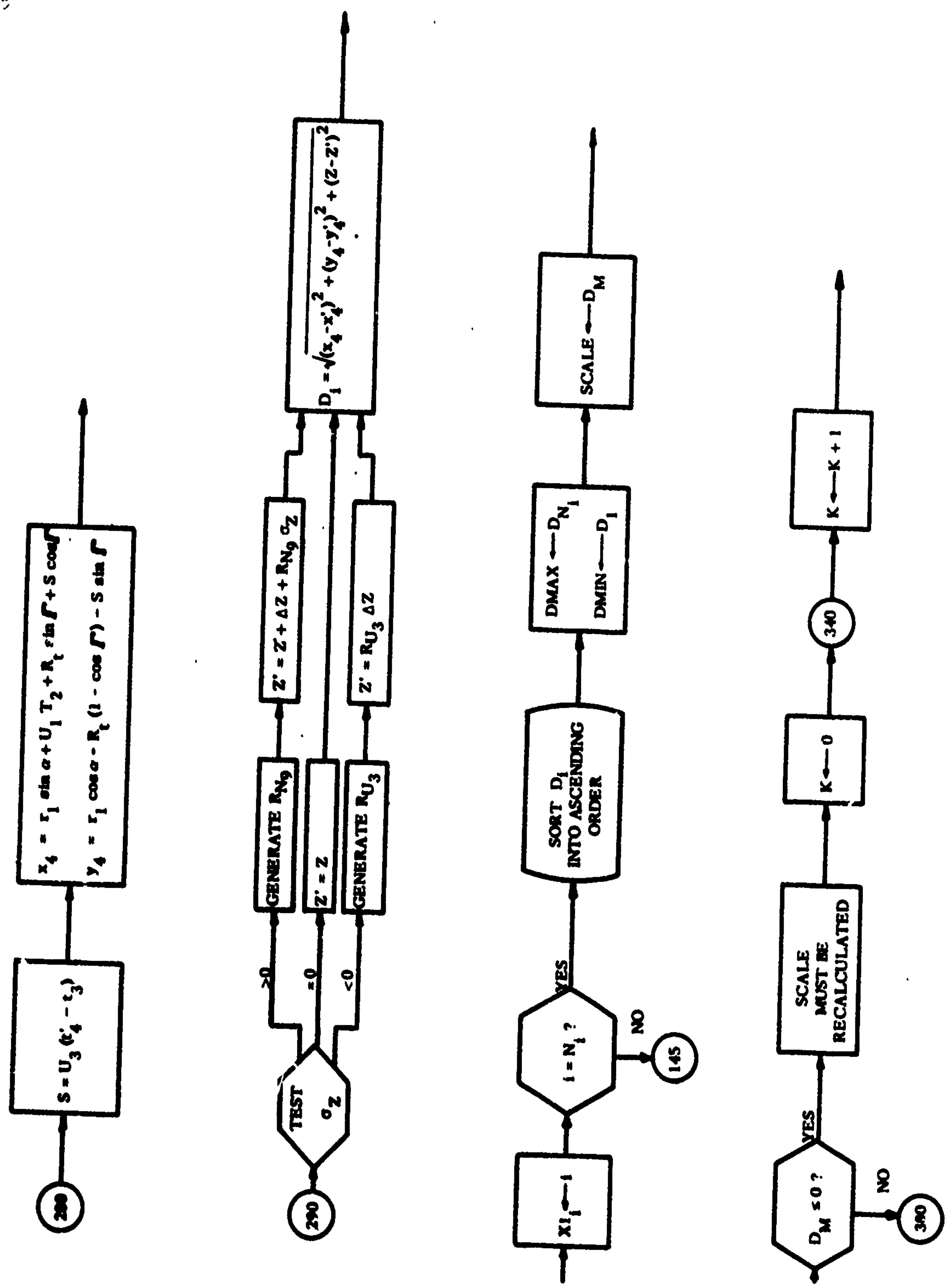
Reference: (a) Operations Evaluation Group, "Design of Antisubmarine Attack Models," in preparation.

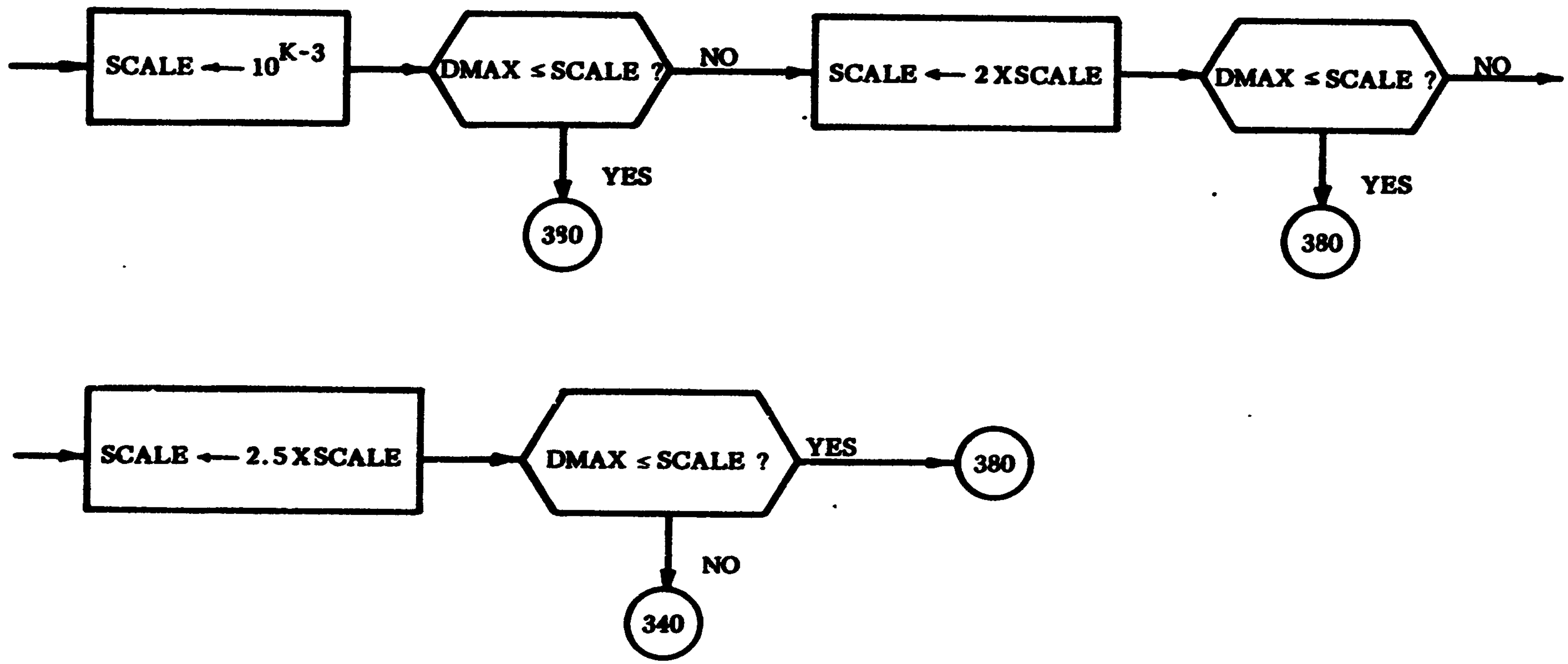
APPENDIX A
FLOW CHARTS



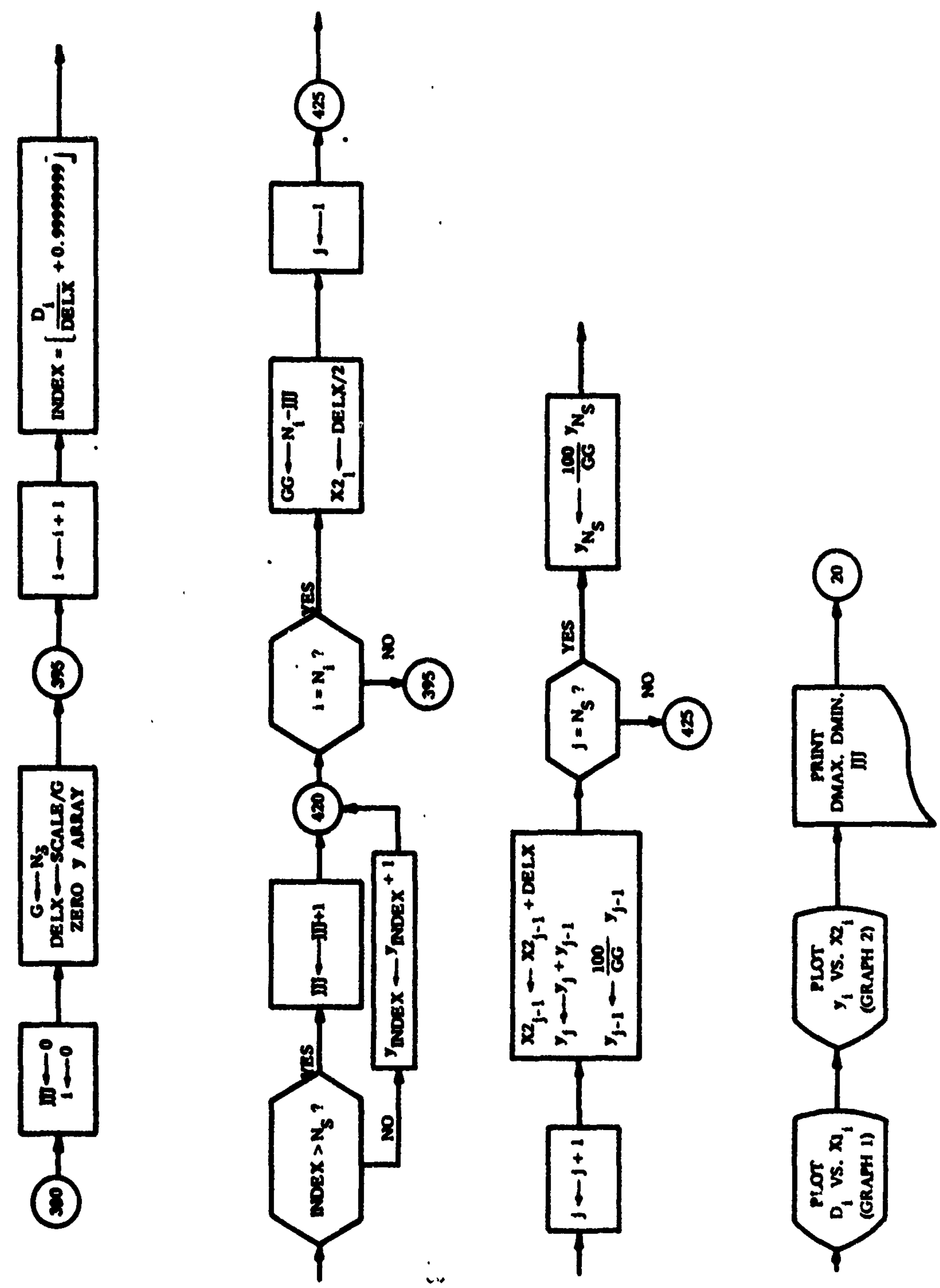
A-1


180
TEST
GENERATE R_{N_7}
$T_2 = t_2$
GENERATE R_{U_1}
210
>0
=0
<0
TEST
GENERATE R_{N_8}
GENERATE R_{U_2}
240
YES
NO
260
290


200
$S = U_3(t'_4 - t_3)$
$x_4 = r_1 \sin\alpha + U_1 T_2 + R_t \sin\Gamma + S \cos\Gamma$
$y_4 = r_1 \cos\alpha - R_t(1 - \cos\Gamma) - S \sin\Gamma$
290
TEST σ_Z
>0
=0
<0
GENERATE R_{N_9}
$Z' = Z$
GENERATE R_{U_3}
$Z' = Z + \Delta Z + R_{N_9}\sigma_Z$
$Z' = R_{U_3}\Delta Z$
$D_i = \sqrt{(x_4 - x'_4)^2 + (y_4 - y'_4)^2 + (Z - Z')^2}$
$XI_i \leftarrow i$
$i = N_i$?
YES
NO
145
SORT D_i INTO ASCENDING ORDER
DMAX $\leftarrow D_{N_i}$
DMIN $\leftarrow D_1$
SCALE $\leftarrow D_M$
$D_M = 0$?
YES
NO
380
SCALE MUST BE RECALCULATED
$K \leftarrow 0$
340
$K \leftarrow K + 1$



SCALE ← 10^{K-3}
DMAX ≤ SCALE ?
NO
YES
330
SCALE ← 2 X SCALE
DMAX ≤ SCALE ?
NO
YES
380
SCALE ← 2.5 X SCALE
DMAX ≤ SCALE ?
YES
380
NO
340



390
III ← 0
I ← 0
G ← N_S
DELX ← SCALE/G
ZERO Y ARRAY
395
I ← I + 1
INDEX = [D_I/DELX + 0.9999999]
INDEX > N_S ?
YES
NO
III ← III+1
Y_INDEX ← Y_INDEX + 1
420
I = N_I ?
YES
NO
395
GG ← N_I − III
X2_1 ← DELX/2
J ← 1
425
J ← J + 1
X2_J-1 ← X2_J-1 + DELX
Y_J ← Y_J + Y_J-1
Y_J-1 ← 100/GG Y_J-1
J = N_S ?
YES
NO
425
Y_NS ← 100/GG Y_NS
PLOT D_I VS. X1_I (GRAPH 1)
PLOT Y_I VS. X2_I (GRAPH 2)
PRINT DMAX, DMIN, III
20

APPENDIX B

FORTRAN STATEMENTS

```
COMPUTER PROGRAM 13-64P     ANTI-SUBMARINE ATTACK AH/SAD
      DIMENSION X1(9000),DD(9000),X2(1000),YY(1000),QQ(36),G1(12),G2(15)
      EQUIVALENCE (QQ(3),DSUBM),(QQ(6),RSUB1),(QQ(7),DELR1),(QQ(8),SIGR1
     X),(QQ(9),RTARN),(QQ(10),ALPHA),(QQ(11),DELALP),(QQ(12),SIGALP),
     X(QQ(13),GAMMA),(QQ(14),SIGGAM),(QQ(15),DELDEL),(QQ(16),SIGDEL),
     X(QQ(17),U1KNOT),(QQ(18),U2KNOT),(QQ(19),U3KNOT),(QQ(20),DELUA),
     X(QQ(21),SIGUA),(QQ(22),TIME2),(QQ(23),SIGT2),(QQ(24),T3P),
     X(QQ(25),ETAP),(QQ(26),AITCHP),(QQ(27),DELERP),(QQ(28),SIGERP),
     X(QQ(29),DELEDP),(QQ(30),SIGEDP),(QQ(31),DELEBP),(QQ(32),SIGEBP),
     X(QQ(33),ZEE),(QQ(34),DELZ),(QQ(35),SIGZ),(QQ(36),VPRIME)
      DO 10 I=1,36
   10 QQ(I)=0.
      KEE=0
      NSUBS=500
      CALL LGCHAR(8,4H SAD)
      PI=3.14159
B       G1(12)=740610306060
B       G1(11)=275121473060
B       G1(10)=016060606060
B       G1(9)=603124601360
B       G1(7)=606060606060
B       G1(6)=443162626024
B       G1(5)=316263214523
B       G1(4)=256260314560
B       G1(3)=216223254524
B       G1(2)=314527604651
B       G1(1)=242551606060
B       G2(15)=741006306060
      G2(14)=G1(11)
B       G2(13)=026060606060
      G2(12)=G1(9)
      G2(10)=G1(7)
B       G2(9)=236444644321
B       G2(8)=633165256026
B       G2(7)=512550642545
B       G2(6)=237060243162
B       G2(5)=635131226463
B       G2(4)=314645604626
B       G2(3)=604431626260
B       G2(2)=243162632145
B       G2(1)=232562606060
   20 CALL DATA(QQ,IND)
      IF(IND)50,50,30
   30 PRINT 40
   40 FORMAT(1H1)
      CALL LGCHAR(8,4H SAD)
      CALL ENDJOB
   50 IDENT=QQ(1)
      NSUBI=QQ(2)
      NSUBS=QQ(4)
      IF(KEE) 52,51,52
   51 NEMPTY=QQ(5)
   52 PRINT 60
   60 FORMAT (1H1, 41H CNA PROGRAM 13-64P ANTI-SUBMARINE ATTACK ///
     X8H ADDRESS, 5X, 5HVALUE, 11X, 11HDESCRIPTION/)
      GEMPTY=NEMPTY
      PRINT 70,(I,QQ(I),I=1,4),GEMPTY,(I,QQ(I),I=6,10)
   70 FORMAT(I5,F15.4,5X,21HIDENTIFICATION NUMBER/
     XI5,F15.4,5X,29HNO. OF MONTE-CARLO ITERATIONS/
```

```
      XI5,F15.4,5X,35HLARGEST VALUE FOR X-AXIS ON GRAPH 2/
      XI5,F15.4,5X,37HNUMBER OF EQUAL INCREMENTS ON GRAPH 2/
      X5H    5,F15.4,5X,33HNUMBER OF EMPTY RANDOM NO. PASSES/
      XI5,F15.4,5X,7HR SUB 1/
      XI5,F15.4,5X,13HDELTA R SUB 1/
      XI5,F15.4,5X,13HSIGMA R SUB 1/
      XI5,F15.4,5X,7HR SUB T/
      XI5,F15.4,5X,5HALPHA/)
       PRINT 80,(I,OO(I),I=11,20)
   80 FORMAT(I5,F15.4,5X,11HDELTA ALPHA/
      XI5,F15.4,5X,15HSIGMA SUB ALPHA/
      XI5,F15.4,5X,5HGAMMA/
      XI5,F15.4,5X,15HSIGMA SUB GAMMA/
      XI5,F15.4,5X,11HDELTA DELTA/
      XI5,F15.4,5X,15HSIGMA SUB DELTA/
      XI5,F15.4,5X,7HU SUB 1/
      XI5,F15.4,5X,7HU SUB 2/
      XI5,F15.4,5X,7HU SUB 3/
      XI5,F15.4,5X,13HDELTA U SUB 1/)
       PRINT 90,(I,OO(I),I=21,30)
   90 FORMAT(I5,F15.4,5X,17HSIGMA SUB U SUB 1/
      XI5,F15.4,5X,7HT SUB 2/
      XI5,F15.4,5X,17HSIGMA SUB T SUB 2/
      XI5,F15.4,5X,13HT PRIME SUB 3/
      XI5,F15.4,5X,9HETA PRIME/
      XI5,F15.4,5X,7HH PRIME/
      XI5,F15.4,5X,25HDELTA EPSILON SUB R PRIME/
      XI5,F15.4,5X,29HSIGMA SUB EPSILON SUB R PRIME/
      XI5,F15.4,5X,25HDELTA EPSILON SUB D PRIME/
      XI5,F15.4,5X,29HSIGMA SUB EPSILON SUB D PRIME/)
       PRINT 100,(I,OO(I),I=31,36)
  100 FORMAT(I5,F15.4,5X,28HDELTA EPSILON SUB BETA PRIME/
      XI5,F15.4,5X,32HSIGMA SUB EPSILON SUB BETA PRIME/
      XI5,F15.4,5X,1HZ/
      XI5,F15.4,5X,7HDELTA Z/
      XI5,F15.4,5X,11HSIGMA SUB Z/
      XI5,F15.4,5X,7HV PRIME////)
       IF(NEMPTY) 130,140,110
  110 DO 120 I=1,NEMPTY
      CALL RANUMH(DUMMY)
  120 CALL GRNUM4(DUMMY)
  130 NEMPTY=0
  140 U1FPS=1.6878065*U1KNOT
      U2FPS=1.6878065*U2KNOT
      U3FPS=1.6878065*U3KNOT
      DELU1=1.6878065*DELUA
      SIGU1=1.6878065*SIGUA
      RTURN=3.*RTARN
      DO 330 I=1,NSUBI
      CALL GRNUM4(RN1)
      CALL GRNUM4(RN2)
      CALL GRNUM4(RN3)
      CALL GRNUM4(RN4)
      U1P=U1FPS+DELU1+RN1*SIGU1
      ALPHAP=ALPHA+DELALP+RN2*SIGALP
      R1P=RSUB1+DELR1+RN3*SIGR1
      DELP=90.+DELDEL+RN4*SIGDEL
      X3P=R1P*SINDF(ALPHAP)+U1P*T3P*SINDF(DELP)+AITCHP*SINDF(ETAP)
```

```
      Y3P=R1P*COSDF(ALPHAP)+U1P*T3P*COSDF(DELP)+AITCHP*COSDF(ETAP)
      CALL GRNUM4(RN5)
      CALL GRNUM4(RN6)
      ERP=DELERP+RN5*SIGERP
      R3P=SQRTF(X3P**2+Y3P**2)
      IF (DELEBP) 170,150,170
  150 IF (SIGEBP) 170,160,170
C     CALCULATIONS FOR OPTION 1
  160 EDP=DELEDP+RN6*SIGEDP
      X4P=X3P+ERP*X3P/R3P+EDP*Y3P/R3P
      Y4P=Y3P+ERP*Y3P/R3P-EDP*X3P/R3P
      R4P=SQRTF(X4P**2+Y4P**2)
      GO TO 180
C     CALCULATIONS FOR OPTION 2
  170 EBP=DELEBP+RN6*SIGEBP
      R4P=R3P+ERP
      ANGLE=ATANDF(Y3P/X3P)
      IF (X3P) 172,174,174
  172 ANGLE=ANGLE+180.
  174 BETAP=90.-ANGLE
      X4P=R4P*SINDF(BETAP+EBP)
      Y4P=R4P*COSDF(BETAP+EBP)
  180 T4P=T3P+(R4P-R3P)/VPRIME
      TT2=TIME2
      IF (SIGT2) 190,210,200
  190 CALL RANUMB (RU1)
      TT2=RU1*TIME2
      GO TO 210
  200 CALL GRNUM4(RN7)
      TT2=TIME2+RN7*SIGT2
  210 GGAMMA=GAMMA
      IF (SIGGAM) 220,240,230
  220 CALL RANUMB (RU2)
      GGAMMA=RU2*GAMMA
      GO TO 240
  230 CALL GRNUM4(RN8)
      GGAMMA=GAMMA+RN8*SIGGAM
  240 IF (T4P-TT2) 250,250,260
  250 XFOUR=RSUB1*SINDF(ALPHA)+U1FPS*T4P
      YFOUR=RSUB1*COSDF(ALPHA)
      GO TO 290
  260 TIME3=TT2+(PI/180.)*GGAMMA*RTURN/U2FPS
      IF(T4P-TIME3) 270,270,280
  270 THETA=((T4P-TT2)/(TIME3-TT2))*GGAMMA
      XFOUR=RSUB1*SINDF(ALPHA)+U1FPS*TT2+RTURN*SINDF(THETA)
      YFOUR=RSUB1*COSDF(ALPHA)-RTURN*(1.-COSDF(THETA))
      GO TO 290
  280 ESS=U3FPS*(T4P-TIME3)
      XFOUR=RSUB1*SINDF(ALPHA)+U1FPS*TT2+RTURN*SINDF(GGAMMA)+ESS*COSDF
     X(GGAMMA)
      YFOUR=RSUB1*COSDF(ALPHA)-RTURN*(1.-COSDF(GGAMMA))-ESS*SINDF
     X(GGAMMA)
  290 ZPRIME=ZEE
      IF (SIGZ) 300,320,310
  300 CALL RANUMB(RU3)
      ZPRIME=RU3*DELZ
      GO TO 320
  310 CALL GRNUM4(RN9)
```

```
      ZPRIME=ZEE+DELZ+RN9*SIGZ
320 DD(I)=SQRTF((XFOUR-X4P)**2+(YFOUR-Y4P)**2+(ZEE-ZPRIME)**2)
330 X1(I)=I
      CALL FORAST(DD,NSUBI,1)
      DMAX=DD(NSUBI)
      SCALE=DSUBM
      IF (DSUBM) 340,340,380
340 DO 370 K=1,41
      SCALE=.001*10.**K
      IF (DMAX-SCALE) 380,380,350
350 SCALE=2.*SCALE
      IF (DMAX-SCALE) 380,380,360
360 SCALE=2.5*SCALE
      IF (DMAX-SCALE) 380,380,370
370 CONTINUE
380 JJJ=0
      GGG=NSUBS
      DELX=SCALE/GGG
      DO 390 I=1,NSUBS
390 YY(I)=0.
      DO 420 I=1,NSUBI
      INDEX=DD(I)/DELX+0.99999999
      IF (INDEX-NSUBS) 410,410,400
400 JJJ=JJJ+1
      GO TO 420
410 YY(INDEX)=YY(INDEX)+1.
420 CONTINUE
      GNORM=NSUBI-JJJ
      X2(1)=DELX/2.
      DO 430 J=2,NSUBS
      X2(J)=X2(J-1)+DELX
      YY(J)=YY(J)+YY(J-1)
430 YY(J-1)=100.*YY(J-1)/GNORM
      YY(NSUBS)=100.*YY(NSUBS)/GNORM
      G1(8)=BINDF(IDENT,6)
      G2(11)=G1(8)
      XUPPER=X1(NSUBI)
      INORM=GNORM
      CALL FNPLOT(G1(12),12H(8H  NUMBER),29H(23H  MISS DISTANCE IN FEET
     X),1.,XUPPER,0.,SCALE,20,20,20,1,6H(F5.0),6H(F6.0))
      CALL CURVE(X1(INORM),DD(INORM),INORM,6H      )
      CALL FNPLOT(G2(15),28H(23H  MISS DISTANCE IN FEET),13H(9H  PERCENT
     X),0.,SCALE,0.,100.,20,2,20,1,6H(F6.0),6H(F4.0))
      CALL CURVE(X2(NSUBS),YY(NSUBS),NSUBS,6H      )
      PRINT 440,DMAX,DD(1),JJJ,SCALE
440 FORMAT(29H THE MAXIMUM MISS DISTANCE = F9.2,5H FEET/29H THE MINIMU
     XM MISS DISTANCE = F9.2,5H FEET//12H THERE WERE I4,30H  MISS DISTAN
     XCES GREATER THAN F9.2,5H FEET)
      KEE=1
      GO TO 20
      END
```

APPENDIX C

SAMPLE PROBLEM SUBMITTAL FORM

OEG COMPUTER DATA SUBMITTAL FORM

Submitted by: J. Doe **Date:** 1 August 1964

Program No. 13-64P **Est. Time** 3 min. **Classification** Uncl

Special Instructions: ——

Address	Value	Address	Value	Address	Value	Address	Value
1	1500	21	3				
2	2000	22	100				
3	0	23	10				
4	100	24	300				
5	5	25	45				
6	5000	26	50				
7	50	27	10				
8	500	28	50				
9	100	29	6				
10	20	30	0				
11	.2	31	3				
12	2	32	2				
13	60	33	50				
14	6	34	.5				
15	.9	35	5				
16	9	26	800				
17	30	——	b ——				
18	20	——	b ——				
19	30						
20	.3						

NOTES:

1. A value of zero must be entered as 0, not left blank.
2. Decimal pts. may be omitted if understood to follow the rightmost digit.
3. The value 3×10^{-5} may be entered as .00003 or 3-5, not as 3×10^{-5}.
4. The factor portion of a value may not contain more than 8 digits.
5. The exponent portion of a value must lie within the range ±39.
6. Exponents may be omitted if zero. If not, they must be signed.
7. Blank cards should be indicated by: ————b————.

APPENDIX D

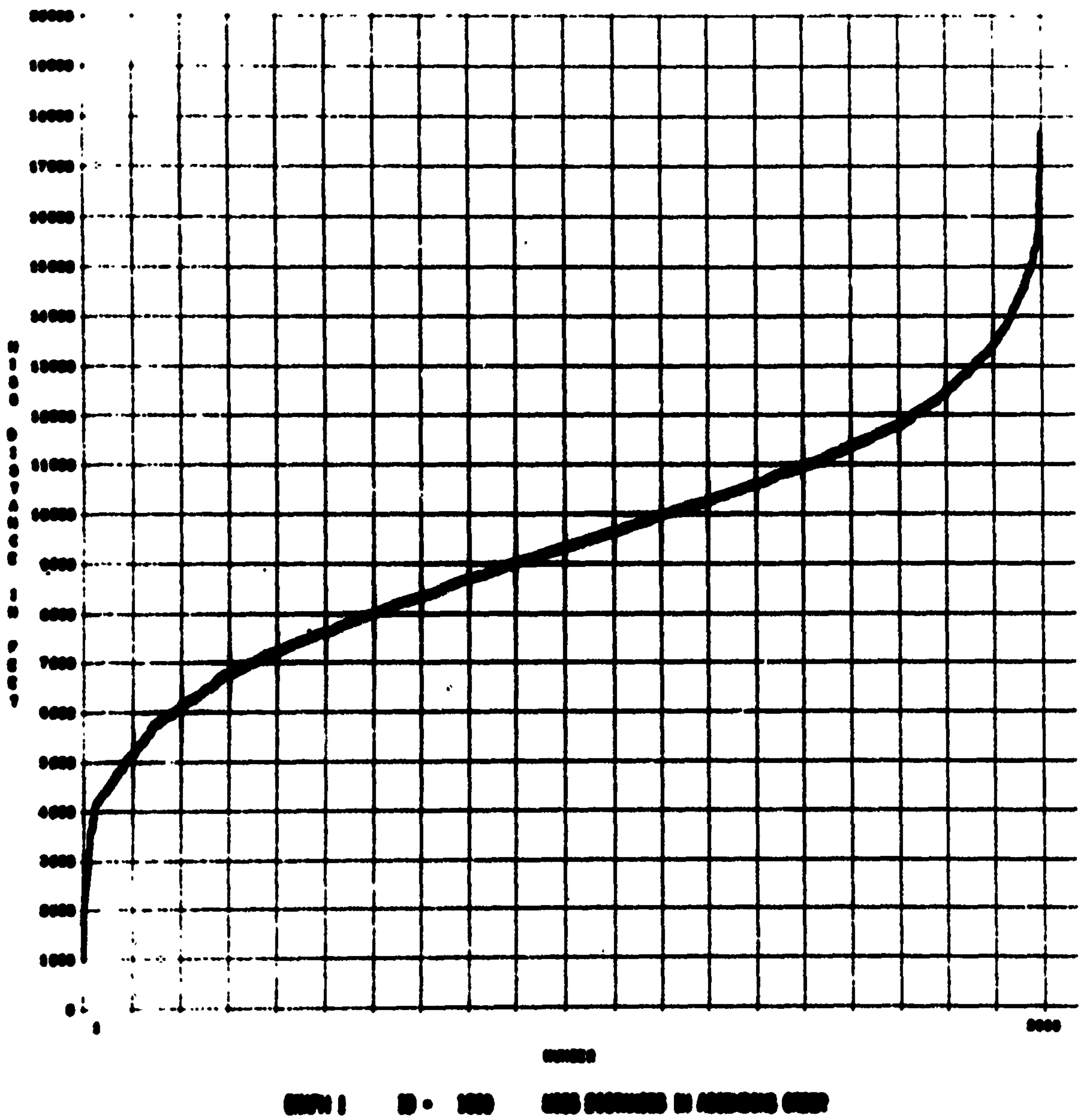
SAMPLE PROBLEM OUTPUT

CNA PROGRAM 13-64P ANTI-SUBMARINE ATTACK

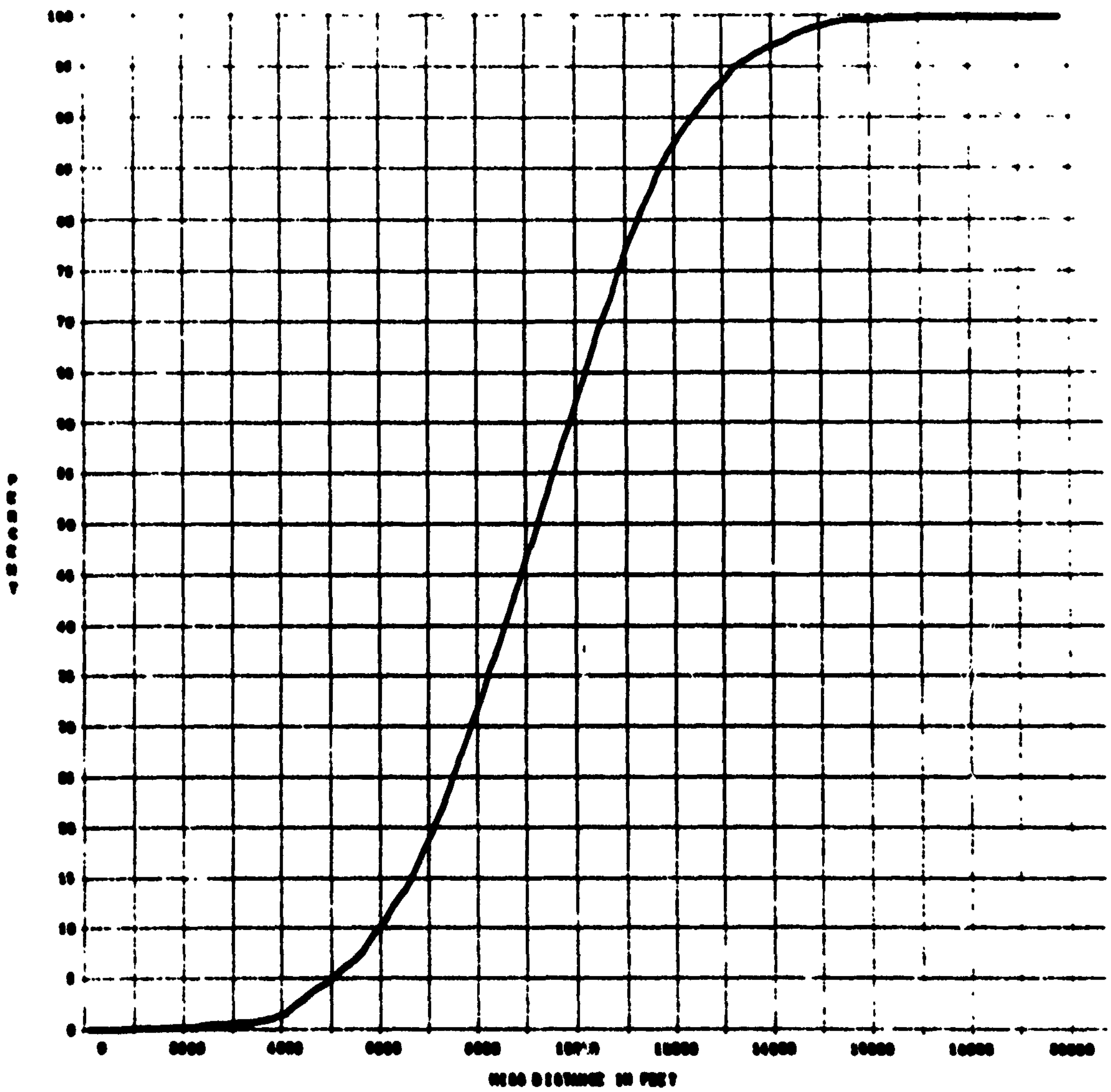
ADDRESS	VALUE	DESCRIPTION
1	1500.0000	IDENTIFICATION NUMBER
2	2000.0000	NO. OF MONTE-CARLO ITERATIONS
3	0.	LARGEST VALUE FOR X-AXIS ON GRAPH 2
4	100.0000	NUMBER OF EQUAL INCREMENTS ON GRAPH 2
5	5.0000	NUMBER OF EMPTY RANDOM NO. PASSES
6	5000.0000	R SUB 1
7	50.0000	DELTA R SUB 1
8	500.0000	SIGMA R SUB 1
9	100.0000	R SUB T
10	20.0000	ALPHA
11	0.2000	DELTA ALPHA
12	2.0000	SIGMA SUB ALPHA
13	60.0000	GAMMA
14	6.0000	SIGMA SUB GAMMA
15	0.9000	DELTA DELTA
16	9.0000	SIGMA SUB DELTA
17	30.0000	U SUB 1
18	20.0000	U SUB 2
19	30.0000	U SUB 3
20	0.3000	DELTA U SUB 1
21	3.0000	SIGMA SUB U SUB 1
22	100.0000	T SUB 2
23	10.0000	SIGMA SUB T SUB 2
24	300.0000	T PRIME SUB 3
25	45.0000	ETA PRIME
26	50.0000	H PRIME
27	10.0000	DELTA EPSILON SUB R PRIME
28	50.0000	SIGMA SUB EPSILON SUB R PRIME
29	0.	DELTA EPSILON SUB D PRIME
30	0.	SIGMA SUB EPSILON SUB D PRIME
31	3.0000	DELTA EPSILON SUB BETA PRIME
32	2.0000	SIGMA SUB EPSILON SUB BETA PRIME
33	50.0000	Z
34	0.5000	DELTA Z
35	5.0000	SIGMA SUB Z
36	800.0000	V PRIME

THE MAXIMUM MISS DISTANCE = 17728.80 FEET
THE MINIMUM MISS DISTANCE = 1012.03 FEET

THERE WERE 0 MISS DISTANCES GREATER THAN 20000.00 FEET



D-2



D-3
(REVERSE BLANK)

APPENDIX E

DATA SUBROUTINE

1. Introduction:

Many computer programs require the flexibility of varying any or all of the parameters in a computer run. Although FORTRAN is fairly flexible in its arithmetic and control statements, its input-output statements are quite rigid. In order to read cards for instance, considerable effort must be expended by the FORTRAN programmer in writing his input statements. This subroutine eliminates some of that tedium. The concept of a "data set" is used. A data set consists of a sequence of punched cards terminated by one blank card. A parameter deck for a computer run may consist of several data sets. Such a parameter deck is terminated by two blank cards.

2. Parameter Addresses:

The primary advantage of this subroutine over FORTRAN input statements results from the use of "parameter addresses." An address is a relative location in the computer memory. It is the subscript of an array or **matrix**. For example, in an array called X, the parameter value X_{53} would be located at address 53. By using the parameter addresses, a user of the program need submit only those parameter values in a data set that are different from those in the previous set.

Three types of addresses are permitted by this subroutine.

(1) A numeric address consisting of one to five characters, each of which is a digit 0 - 9. Such an address (n) refers to the n^{th} element in a specified array.

(2) An alpha address consisting of one to six characters, the first of which must be alphabetic (A-Z). The remaining may be alphabetic or numeric (A-Z or 0-9). Such an address refers to the n^{th} element in a specified array ($1 \leq n \leq 26$), where the first character of the address corresponds to n as the 26 letters of the alphabet correspond to the integers 1-26.

(3) A matrix address consisting of two or more numeric fields separated by commas. For example, the address 53, 47 refers to the element in the 53^{rd} row and the 47^{th} column of a two-dimensional matrix. There is no limit to the number of dimensions in a matrix address.

3. Input Card Format:

A standard submittal form (see attachment) has been designed for the analyst. This form provides for entering parameter values with their associated addresses. The user indicates blank cards to separate data sets. The keypunch operator has the option of punching one address and value per card, or, if the addresses are sequential, of punching one address and several values on a card.

Only columns 1-72 of a card are used. Each column must contain one of the following: a digit (0-9), a "+" or " - " sign or a dash, a letter (A-Z), a period, a comma, or a blank. Each punched card must contain one parameter address. The address may start in column 1, or, if desired, may start in a later column, provided all columns before it are blank. The address is terminated by at least one blank column. Only one address is permitted on the card. Succeeding columns contain one or more parameter values, each separated by one or more blank columns. A value may be signed or unsigned. The length of the value field is variable. No blanks are permitted within a value field. A value may be punched with or without an exponent. An exponent is recognized by the presence of a plus or minus sign (or dash) between the fractional part and exponent part of the value. Decimal points (periods) may be punched in either the fractional or exponent parts of a value. If more than one value is punched on a card, those after the first will be entered at sequential addresses relative to the address of the first value.

4. Usage:

A data set is read by the use of the statement:

CALL DATA (X, I)

in a FORTRAN program for the IBM 7090. The argument X is the name of an array in the program. The argument I is an indicator set by the subroutine. This indicator may be tested by the main program upon return from the subroutine. It will have a value of 0 or 1 or 2.

0: The subroutine has read a data set. The main program will normally proceed to operate on this data.

1: The subroutine has read the second blank card which terminates the parameter deck. The main program will normally terminate at this point.

2: The subroutine has read a "bad" data card. The main program may terminate the run, or ignore the card and return to the subroutine to read the rest of the data set.

If the cards to be read contain matrix addresses, additional arguments must be included in the FORTRAN calling statement:

$$\text{CALL DATA } (X, D_1, D_2, D_3, \ldots, D_n, I)$$

where D_i is the i^{th} dimension of the matrix X.

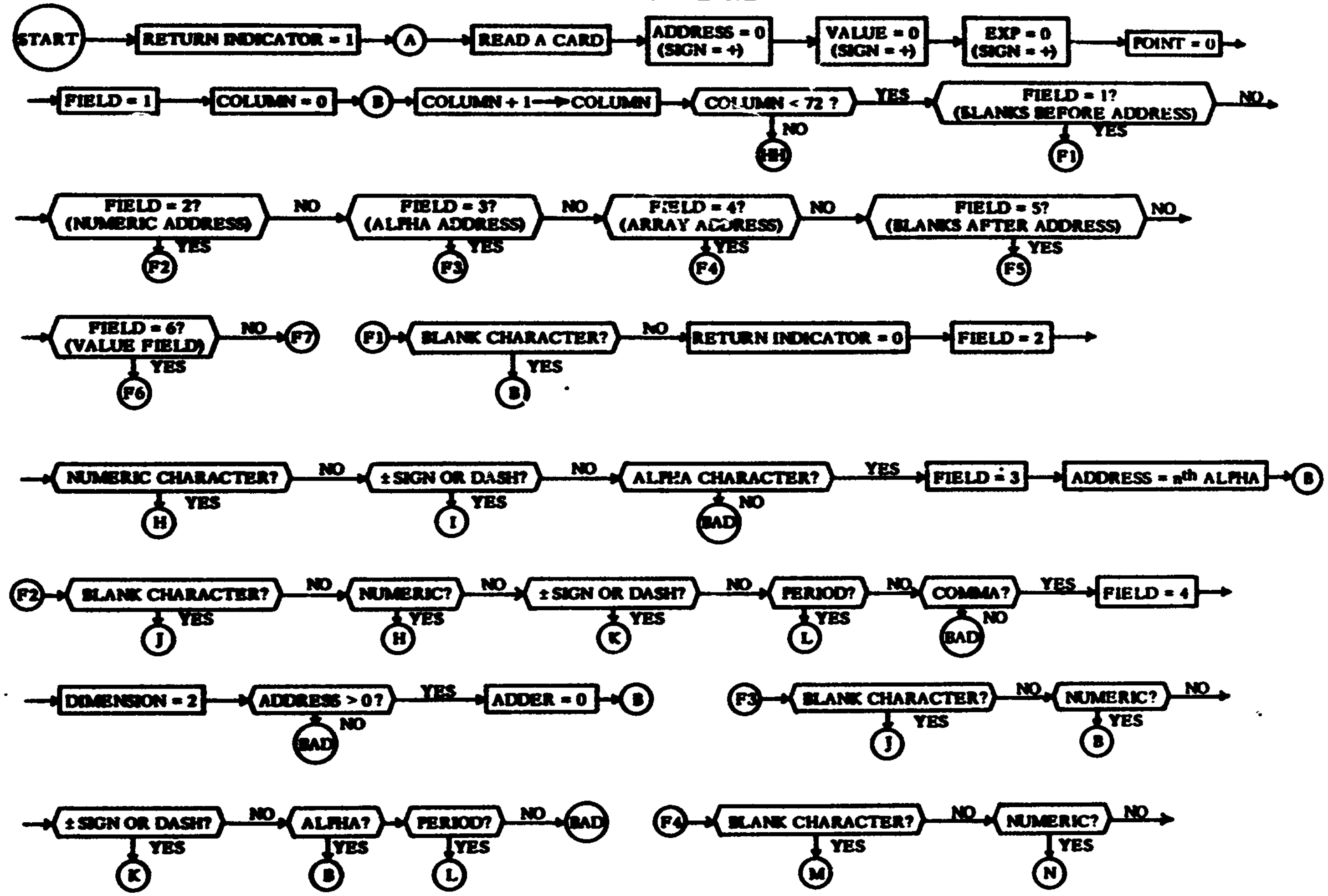
5. Method:

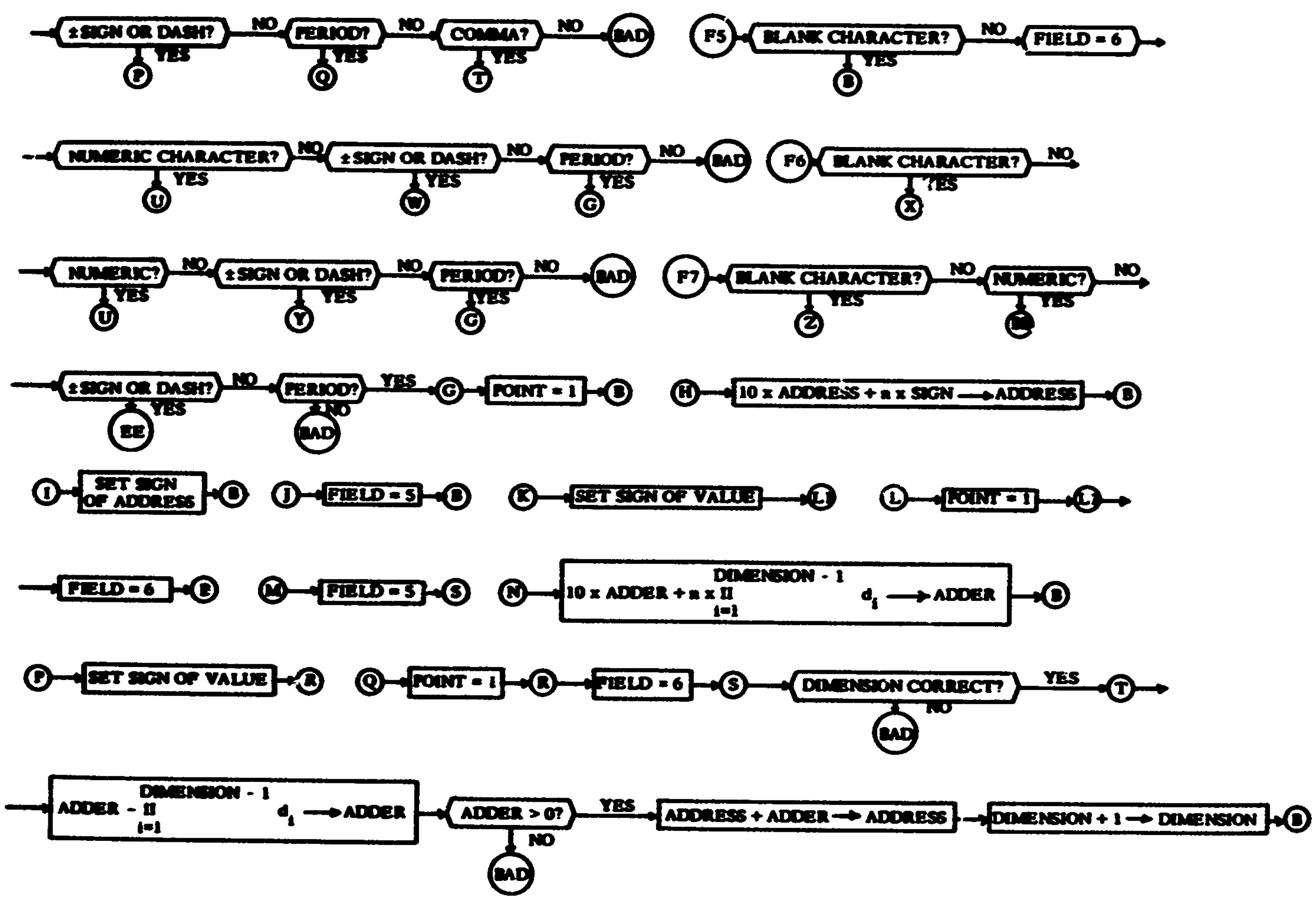
See the attached flow chart. DATA reads parameter values and loading addresses from cards. If sense switch 5 is up, it will read the values and addresses from tape (unit A2). It converts the values to floating point numbers, and stores them as elements of an array specified in the calling statement. The elements are specified by the addresses. If a card (or tape record) is read which contains non-permitted characters (see input card format above), DATA prints the statement "bad data card," followed by an image of the card itself.

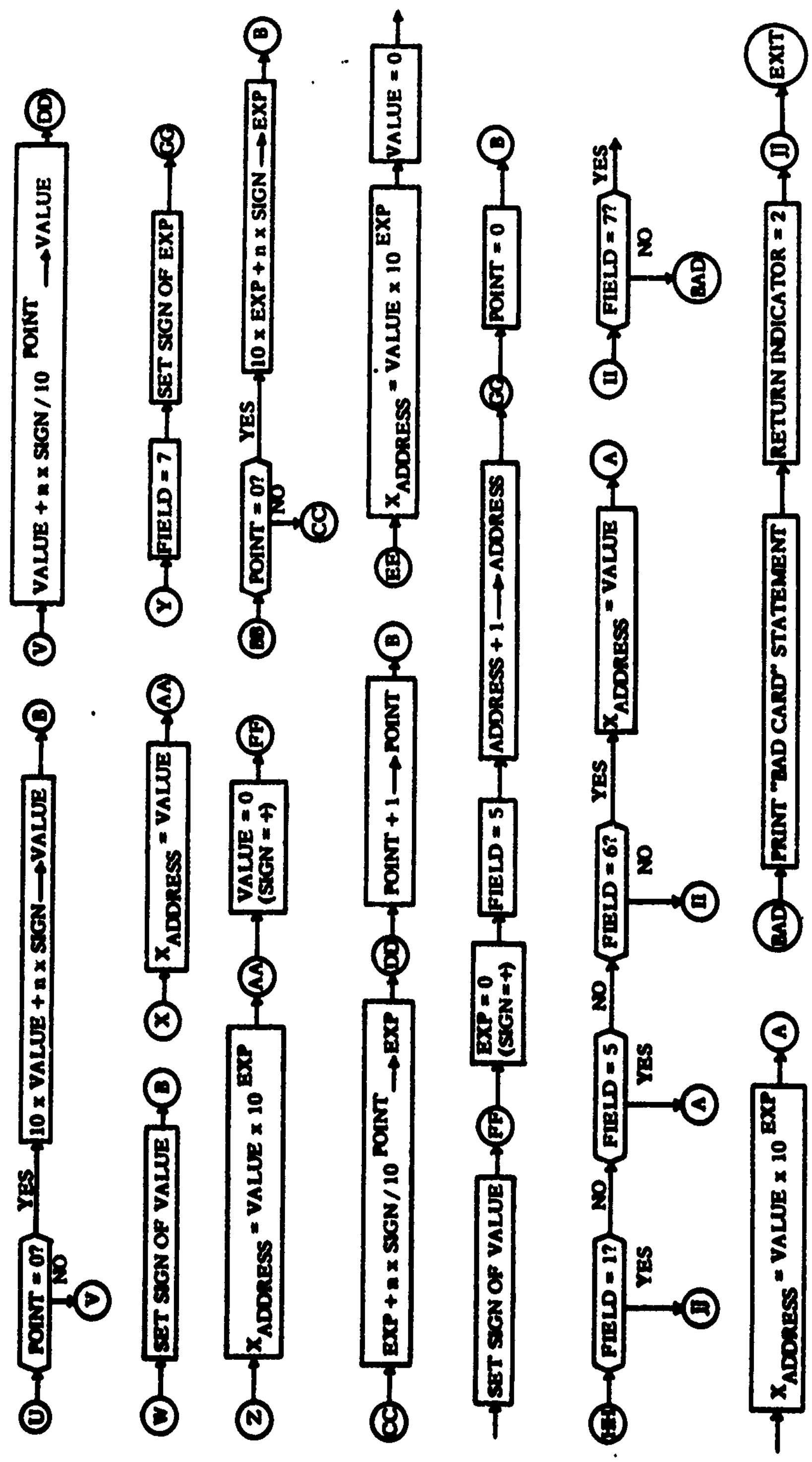
6. <u>Coding Information:</u>

See the symbolic listing included in this appendix. DATA is written in the 7090 FAP language. It must be used in conjunction with the BELL system. It requires 401 words storage space.

SUBROUTINE "DATA"




± SIGN OR DASH?
YES
NO
PERIOD?
YES
NO
COMMA?
YES
NO
BAD
F5
BLANK CHARACTER?
YES
NO
FIELD = 6
NUMERIC CHARACTER?
YES
NO
± SIGN OR DASH?
YES
NO
PERIOD?
YES
NO
BAD
F6
BLANK CHARACTER?
YES
NO
NUMERIC?
YES
NO
± SIGN OR DASH?
YES
NO
PERIOD?
YES
NO
BAD
F7
BLANK CHARACTER?
YES
NO
NUMERIC?
YES
NO
± SIGN OR DASH?
YES
EE
NO
PERIOD?
NO
BAD
YES
POINT = 1
10 x ADDRESS + n x SIGN ⟶ ADDRESS
SET SIGN OF ADDRESS
FIELD = 5
SET SIGN OF VALUE
POINT = 1
FIELD = 6
FIELD = 5
10 x ADDER + n x $\prod_{i=1}^{DIMENSION-1} d_i$ ⟶ ADDER
SET SIGN OF VALUE
POINT = 1
FIELD = 6
DIMENSION CORRECT?
NO
BAD
YES
ADDER - $\prod_{i=1}^{DIMENSION-1} d_i$ ⟶ ADDER
ADDER > 0?
NO
BAD
YES
ADDRESS + ADDER ⟶ ADDRESS
DIMENSION + 1 ⟶ DIMENSION


U
POINT = 0?
NO
V
YES
10 x VALUE + n x SIGN → VALUE
B
V
VALUE + n x SIGN / 10 POINT → VALUE
DD
W
SET SIGN OF VALUE
B
X
X ADDRESS = VALUE
AA
Y
FIELD = 7
SET SIGN OF EXP
GG
Z
X ADDRESS = VALUE x 10 EXP
AA
VALUE = 0 (SIGN = +)
FF
BB
POINT = 0?
NO
CC
YES
10 x EXP + n x SIGN → EXP
B
CC
EXP + n x SIGN / 10 POINT → EXP
DD
POINT + 1 → POINT
B
EE
X ADDRESS = VALUE x 10 EXP
VALUE = 0
SET SIGN OF VALUE
FF
EXP = 0 (SIGN = +)
FIELD = 5
ADDRESS + 1 → ADDRESS
GG
POINT = 0
B
HH
FIELD = 1?
YES
JJ
NO
FIELD = 5
YES
A
NO
FIELD = 6?
NO
II
YES
X ADDRESS = VALUE
A
II
FIELD = 7?
NO
BAD
YES
X ADDRESS = VALUE x 10 EXP
A
BAD
PRINT "BAD CARD" STATEMENT
RETURN INDICATOR = 2
JJ
EXIT

E-6

SYMBOLIC LISTING

```
        FAP
        ENTRY DATA
DATA    SXA  X1,1
        SXA  X2,2
        SXA  X4,4
        CAL  1,4
        ADD  CORE
        STO  XLOC
        AXT  1,1
        SXA  *+1,1
        CAL  **,4
        ANA  MASK
        TNZ  *+2
        TXI  *-4,1,1
        SXA  EXIT,1
        TXI  *+1,1,-1
        SXA  *+1,1
        CLA  **,4
        STA  A1
        STA  F1A
        STA  II2
        AXT  1,1             RETURN INDICATOR = 1
A1      SXD  **,1
A       TSX  HHREAD,4        READ A CARD
        PZE  CARD
        TRA  EXIT
        TRA  BAD
        STZ  ADDRES          ADDRESS = 0
        STZ  VALUE           VALUE = 0
        STZ  EXP             EXP = 0
        STZ  POINT           POINT = 0
        AXT  1,1             FIELD = 1
        SXA  FIELD,1
        AXT  13,1
A2      TNX  HH,1,1          COLUMN GT 72
        AXT  42,2
        SXA  COLUMN,2
B       LXA  COLUMN,2        COLUMN = COLUMN+1
        TNX  A2,2,6
        SXA  COLUMN,2
        LDQ  CARD+12,1
        RQL  36,2
        PXD  0,0
        LGL  6
        STO  CHARAC
        ORA  FLOAT
        FAD  FLOAT
        STO  NUMB
        AXT  42,4
        CLA  CHARAC
        CAS  TABLE+42,4
        TRA  *+2
        TRA  *+3
        TIX  *-3,4,1
```

```
         TRA  BAD
         LXA  FIELD,2
         TRA  F1+1,2
         TRA  F7              FIELD=7 (EXPONENT FIELD)
         TRA  F6              FIELD=6 (VALUE FIELD)
         TRA  F5              FIELD=5 (BLANKS AFTER ADDRESS)
         TRA  F4              FIELD=4 (ARRAY ADDRESS)
         TRA  F3              FIELD=3 (ALPHA ADDRESS)
         TRA  F2              FIELD=2 (NUMERIC ADDRESS)
F1       TXH  B,4,41          FIELD=1 (BLANKS BEFORE ADDRESS)
F1A      STZ  **              RETURN INDICATOR = 0
         AXT  2,2             FIELD = 2
         SXA  FIELD,2
         TXH  H,4,31          NUMERIC CHARACTER
         TXH  I,4,28          SIGN OR DASH
         TXL  BAD,4,2
         AXT  3,2             ALPHA CHARACTER, FIELD = 3
         SXA  FIELD,2
         TXI  *+1,4,-2        ADDRESS = NTH ALPHA
         SXA  ADDRES,4
         TRA  B
F2       TXH  J,4,41          BLANK CHARACTER
         TXH  H,4,31          NUMERIC CHARACTER
         TXH  K,4,28          SIGN OR DASH
         TXH  BAD,4,2
         TXH  L,4,1           PERIOD
         AXT  4,2             COMMA,FIELD = 4
         SXA  FIELD,2
         AXT  2,2
         SXA  DIMENS,2        DIMENSION = 2
         CLA  ADDRES          TEST ADDRESS
         TZE  BAD
         TMI  BAD
F2A      STZ  ADDER           ADDER=0
         TRA  B
F3       TXH  J,4,41          BLANK CHARACTER
         TXH  B,4,31          NUMERIC CHARACTER
         TXH  K,4,28          SIGN OR DASH
         TXH  B,4,2           ALPHA CHARACTER
         TXH  L,4,1           PERIOD
         TRA  BAD
F4       TXH  M,4,41          BLANK CHARACTER
         TXH  N,4,31          NUMERIC CHARACTER
         TXH  P,4,28          SIGN OR DASH
         TXH  BAD,4,2
         TXH  Q,4,1           PERIOD
         TRA  T               COMMA
F5       TXH  B,4,41          BLANK CHARACTER
         AXT  6,2             FIELD = 6
         SXA  FIELD,2
         TXH  U,4,31          NUMERIC CHARACTER
         TXH  W,4,28          SIGN OR DASH
         TXH  BAD,4,2
         TXH  G,4,1           PERIOD
```

```
         TRA BAD
F6       TXH X,4,41          BLANK CHARACTER
         TXH U,4,31          NUMERIC CHARACTER
         TXH Y,4,28          SIGN OR DASH
         TXH BAD,4,2
         TXH G,4,1           PERIOD
         TRA BAD
F7       TXH Z,4,41          BLANK CHARACTER
         TXH BB,4,31         NUMERIC CHARACTER
         TXH EE,4,28         SIGN OR DASH
         TXH BAD,4,2
         TXL BAD,4,1
G        AXT 1,2             PERIOD, POINT = 1
         SXA POINT,2
         TRA B
H        LDQ ADDRES          ADDRESS = 10 X ADDRESS + N
         MPY H10
         XCA
         ACL CHARAC
         STO ADDRES
         TRA B
I        TXH B,4,30          + SIGN
         CLA ADDRES          SET SIGN OF ADDRESS
         SSM
         STO ADDRES
         TRA B
J        AXT 5,2             FIELD = 5
         SXA FIELD,2
         TRA B
K        TXH L1,4,30         + SIGN
         CLA VALUE           SET SIGN OF VALUE
         SSM
         STO VALUE
         TRA L1
L        AXT 1,2             POINT = 1
         SXA POINT,2
L1       AXT 6,2             FIELD = 6
         SXA FIELD,2
         TRA B
M        AXT 5,2             FIELD = 5
         SXA FIELD,2
         TRA S
N        LDQ ADDER           ADDER = 10 X ADDER + N X PROD
         MPY H10
         STQ ADDER
         TSX T1,4
         MPY CHARAC
         XCA
         ADD ADDER
         STO ADDER
         TRA B
P        TXH R,4,30          + SIGN
         CLA VALUE           SET SIGN OF VALUE
         SSM
```

```
        STO  VALUE
        TRA  R
Q       AXT  1,2            POINT = 1
        SXA  POINT,2
R       AXT  6,2            FIELD = 6
        SXA  FIELD,2
S       LXA  EXIT,2         CHECK DIMENSION
        TXI  *+1,2,-3
        PXA  0,2
        SUB  DIMENS
        TNZ  BAD
T       TSX  T1,4           ADDER=ADDER-PROD
        CLA  ADDER
        SUB  PROD
        STO  ADDER
        TZE  BAD            CHECK ADDER
        TMI  BAD
        ADD  ADDRES
        STO  ADDRES
        CLA  DIMENS
        ADD  H1
        STO  DIMENS
        TRA  F2A
T1      SXA  T4,4           PROD = PRODUCT OF DIMENSIONS
        CLA  H1
        STO  PROD
        STA  T3
        LXA  DIMENS,2
        TXI  *+1,2,-1
        LXA  X4,4
T2      CAL  T3
        ADD  H1
        STA  T3
T3      CLA  **,4
        STA  *+1
        LDQ  **
        RQL  18
        MPY  PROD
        STQ  PROD
        TIX  T2,2,1
T4      AXT  **,4
        TRA  1,4
U       CLA  POINT          TEST POINT
        TNZ  V
        LDQ  VALUE          VALUE = 10 X VALUE + N
        FMP  DEC10
        SSP
        FAD  NUMB
        LDQ  VALUE
        LLS  0
        STO  VALUE
        TRA  B
V       LXA  POINT,4        VALUE = VALUE + N/(10**POINT)
        CLA  NUMB
```

E-10

```
       FDP DEC10
       XCA
       TIX *-2,4,1
       LDQ VALUE
       LLS 0
       FAD VALUE
       STO VALUE
       TRA DD
W      TXH B,4,30         + SIGN
       CLA VALUE          SET SIGN OF VALUE
       SSM
       STO VALUE
       TRA B
X      CLA XLOC           X(ADDRESS) = VALUE
       SUB ADDRES
       STA *+2
       CLA VALUE
       STO **
       TRA AA
Y      AXT 7,2
       SXA FIELD,2        FIELD = 2
       TXH GG,4,30        + SIGN
       CLA EXP            SET SIGN OF EXP
       SSM
       STO EXP
       TRA GG
Z      CLA XLOC           X(ADDRESS) = VALUE X 10**EXP
       SUB ADDRES
       STA Z1
       CLA DEC10
       LDQ EXP
       CALL EXP(3
       XCA
       FMP VALUE
Z1     STO **
AA     STZ VALUE          VALUE = 0
       TRA FF
BB     CLA POINT          TEST POINT
       TNZ CC
       LDQ EXP            EXP = 10 X EXP + N
       FMP DEC10
       SSP
       FAD NUMB
       LDQ EXP
       LLS 0
       STO EXP
       TRA B
CC     LXA POINT,4        EXP = EXP + N/(10**POINT)
       CLA NUMB
       FDP DEC10
       XCA
       TIX *-2,4,1
       LDQ EXP
       LLS 0
```

E-11

```
         FAD EXP
         STO EXP
DD       CLA POINT        POINT = POINT + 1
         ADD H1
         STO POINT
         TRA B
EE       CLA XLOC         X(ADDRESS) = VALUE X 10**EXP
         SUB ADDRES
         STA EE1
         CLA DEC10
         LDQ EXP
         CALL EXP(3
         XCA
         FMP VALUE
EE1      STO **
         PXD 0,0          VALUE = 0
         TXH *+2,4,30     + SIGN
         SSM              SET SIGN OF VALUE
         STO VALUE
FF       STZ EXP          EXP = 0
         AXT 5,2          FIELD = 5
         SXA FIELD,2
         CAL ADDRES       ADDRESS = ADDRESS + 1
         ADD H1
         SLW ADDRES
GG       STZ POINT        POINT = 0
         TRA B
HH       LXA FIELD,1
         TXL JJ,1,1       FIELD=1, EXIT
         TXL BAD,1,4
         TXL A,1,5        FIELD=5, READ ANOTHER CARD
         TXH II,1,6
         CLA XLOC         FIELD=6, X(ADDRESS) = VALUE
         SUB ADDRES
         STA *+2
         CLA VALUE
         STO **
         TRA A
II       TXH BAD,1,7
         CLA XLOC         FIELD=7,
         SUB ADDRES       X(ADDRESS) = VALUE X 10**EXP
         STA III
         CLA DEC10
         LDQ EXP
         CALL EXP(3
         XCA
         FMP VALUE
III      STO **
         TRA A
BAD      TSX HPRINT,4
         PZE PRINT,0,15
         AXT 2,1
II2      SXD **,1
X1       AXT **,1
```

```
X2     AXT **,2
X4     AXT **,4
EXIT   TRA **,4
MASK   OCT 777777700000
PRINT  BCD 3 BAD DATA CARD...
CARD   BSS 12
ADDRES HTR **
VALUE  HTR **
EXP    HTR **
POINT  HTR **
FIELD  HTR **
COLUMN HTR **
TABLE  OCT 60             BLANK
       OCT 0              0
       OCT 1              1
       OCT 2              2
       OCT 3              3
       OCT 4              4
       OCT 5              5
       OCT 6              6
       OCT 7              7
       OCT 10             8
       OCT 11             9
       OCT 20             + SIGN
       OCT 40             - SIGN
       OCT 14             DASH
       OCT 71             Z
       OCT 70             Y
       OCT 67             X
       OCT 66             W
       OCT 65             V
       OCT 64             U
       OCT 63             T
       OCT 62             S
       OCT 51             R
       OCT 50             Q
       OCT 47             P
       OCT 46             O
       OCT 45             N
       OCT 44             M
       OCT 43             L
       OCT 42             K
       OCT 41             J
       OCT 31             I
       OCT 30             H
       OCT 27             G
       OCT 26             F
       OCT 25             E
       OCT 24             D
       OCT 23             C
       OCT 22             B
       OCT 21             A
       OCT 33             PERIOD
       OCT 73             COMMA
```

```
CHARAC HTR **
DIMENS HTR **
 ADDER HTR **
 H10   HTR 10
 DEC10 DEC 10.0
 H1    HTR 1
 PROD  HTR **
 AMASK OCT 77777
 FLOAT OCT 233000000000
 NUMB  HTR **
 XLOC  HTR **
 CORE  OCT 100001
 JJ    SYN X1
       END
```

OEG COMPUTER DATA SUBMITTAL FORM

Submitted by: ______________________________ **Date:** ______________

Program No. ____________ **Est. Time** ____________ **Classification** ____________

Special Instructions: ____

Address	Value	Address	Value	Address	Value	Address	Value

NOTES:

1. A value of zero must be entered as 0, not left blank.
2. Decimal pts. may be omitted if understood to follow the rightmost digit.
3. The value 3×10^{-5} may be entered as .00003 or 3-5, not as 3×10^{-5}.
4. The factor portion of a value may not contain more than 8 digits.
5. The exponent portion of a value must lie within the range ±39.
6. Exponents may be omitted if zero. If not, they must be signed.
7. Blank cards should be indicated by: ———————b——————— .